

HTML5



Weekly plan (HTML5, 1st semester 2021)

- **wk01** : Introduction to curriculum, current state of HTML5 & github
- **wk02** : Making HTML5 documents
- **wk03** : iFrame, Media, my 1st Homepage & Intro to Semantic tags
- **wk04** : **Semantic tags & Web forms**
- **wk05** : **CSS3 I. Basic & Box model**
- **wk06** : **CSS3 II. Advanced & Animation**
- **wk07** : DIY, **Quiz-15**, **my 2nd Homepage**
- **wk08** : Mid-term Exam. **my 2nd Homepage**
- **wk09** : **JS I. Data types & operators (ch.6)**
- **wk10** : **JS II. if, loop & functions (ch.6)**
- **wk11** : **JS III. Core objects & array (ch.7)**
- **wk12** : **JS IV. DOM (ch.8) & BOM (ch.10)**
- **wk13** : **JS V. Events (ch.9)**
- **wk14** : **my final Homepage (hmn_rpt03)**
- **wk15** : Final exam.

이미지 프리로딩과 정보 출력

이미지를 클릭하면 다음 이미지와 정보를 보여줍니다.



My ID (HTML5, 1st semester 2021)

HM01	김준수
HM02	김현서
HM03	박영훈
HM04	박윤희
HM05	성은지
HM06	손윤우
HM07	신승철
HM08	오세운
HM09	오세현
HM10	우승철

HM11	윤현석
HM12	이예주
HM13	최민석
HM14	강지환
HM15	성인제
HM16	고태승
HM17	김성환
HM18	이정호
HM19	장원일
HM20	장태호
HM21	정지원
HM22	진우태

[DIY] 자바스크립트 DOM, BOM을 활용하는 두 개의 html 문서를 만드시오.

1. [예제 8-4 this 활용 → 예제 8-6 참고]
아래에 “Go back!” 버튼을 추가하고 클릭하면
모든 버튼을 초기 모양으로 되돌리는 html 문서를
작성하시오.

파일명: [hmn_this.html](#)

this 활용

버튼

버튼

여기 클릭하면 크기와 색 변경

Go back!

2. [예제 10-9 location 활용]

“hmn favorite 3 XX” 라는 제목으로 영화, 여행, 노래, 취미 중 하나의 주제를 선정 하시오.
선택된 주제에 대한 Best 3 정보를 select tag에 담아 그 중 하나를 선택해서 보여주는 html
문서를 작성하시오.

파일명: [hmn_best3.html](#)

가점: Javascript 프로그래밍 응용 능력.

Possible result ...

5

this 활용



여기 클릭하면 크기와 색 변경

Go back!



this 활용



여기 클릭하면 크기와 색 변경

Go back!

```
<script>
  function change(obj, size, color) {
    obj.style.color = color;
    obj.style.fontSize = size;
  }

  function reset() {
    var resetArray = document.getElementsByClassName("button");
    for (var i = 0; i < resetArray.length; i++) {
      var reset = resetArray[i];
      reset.style.color = null;
      reset.style.fontSize = null;
    }
  }
</script>
```

```
<button class="button" onclick="change(this, '30px', 'red')">
  버튼
</button>
<button class="button" onclick="change(this, '30px', 'blue')">
  버튼
</button>
<div class="button" onclick="change(this, '25px', 'orange')">
  여기 클릭하면 크기와 색 변경
</div>
<button onclick="reset()">
  go back
</button>
```

Good report

6

this 활용하여 과제 해결



여기 클릭하면 크기와 색 변경

GoBack_btn!!



this 활용하여 과제 해결



여기 클릭하면 크기와 색 변경

GoBack_btn!!

```
<script>
function change(obj, size, color) {
    obj.style.color = color;
    obj.style.fontSize = size;
}

function reset() {
    var resetArray = document.getElementsByClassName("button");
    for (var i = 0; i < resetArray.length; i++) {
        var reset = resetArray[i];
        reset.style.color = null;
        reset.style.fontSize = null;
    }
}
</script>
```

```
<button class="button" onclick="change(this, '30px', 'red')">
    버튼
</button>
<button class="button" onclick="change(this, '30px', 'blue')">
    버튼
</button>
<div class="button" onclick="change(this, '25px', 'orange')">
    여기 클릭하면 크기와 색 변경
</div>
<button onclick="reset()">
    go back
</button>
```

wk12-실습 : 결과를 나의 github에 올리기

7

실습 결과를 github에 올립니다.

1. README.md에는 실습 결과 요약 추가 입력
2. hmxx_this.html 완성
3. hmxx_best3.html 완성
4. "hmxx" repo의 wk12 폴더에 upload

단 업로드가 안될 경우, wk12.zip을 업로드
그리고 집에서 wk12 폴더로 다시 업로드.



09

강의 목표 (9장)

1. 이벤트가 무엇이고 언제 발생하는지 안다.
2. 자바스크립트 코드로 이벤트 리스너를 작성할 수 있다.
3. 발생하는 이벤트가 DOM 트리를 따라 흘러가는 경로를 안다.
4. 마우스 관련 이벤트를 다룰 수 있다.
5. 문서와 이미지의 로딩 완료 시 호출되는 onload 리스너를 작성할 수 있다.
6. 폼에 발생하는 이벤트 리스너를 다룰 수 있다.
7. 키 관련 이벤트를 다룰 수 있다.
8. [DIY] Handling events and image pre-loading

Event

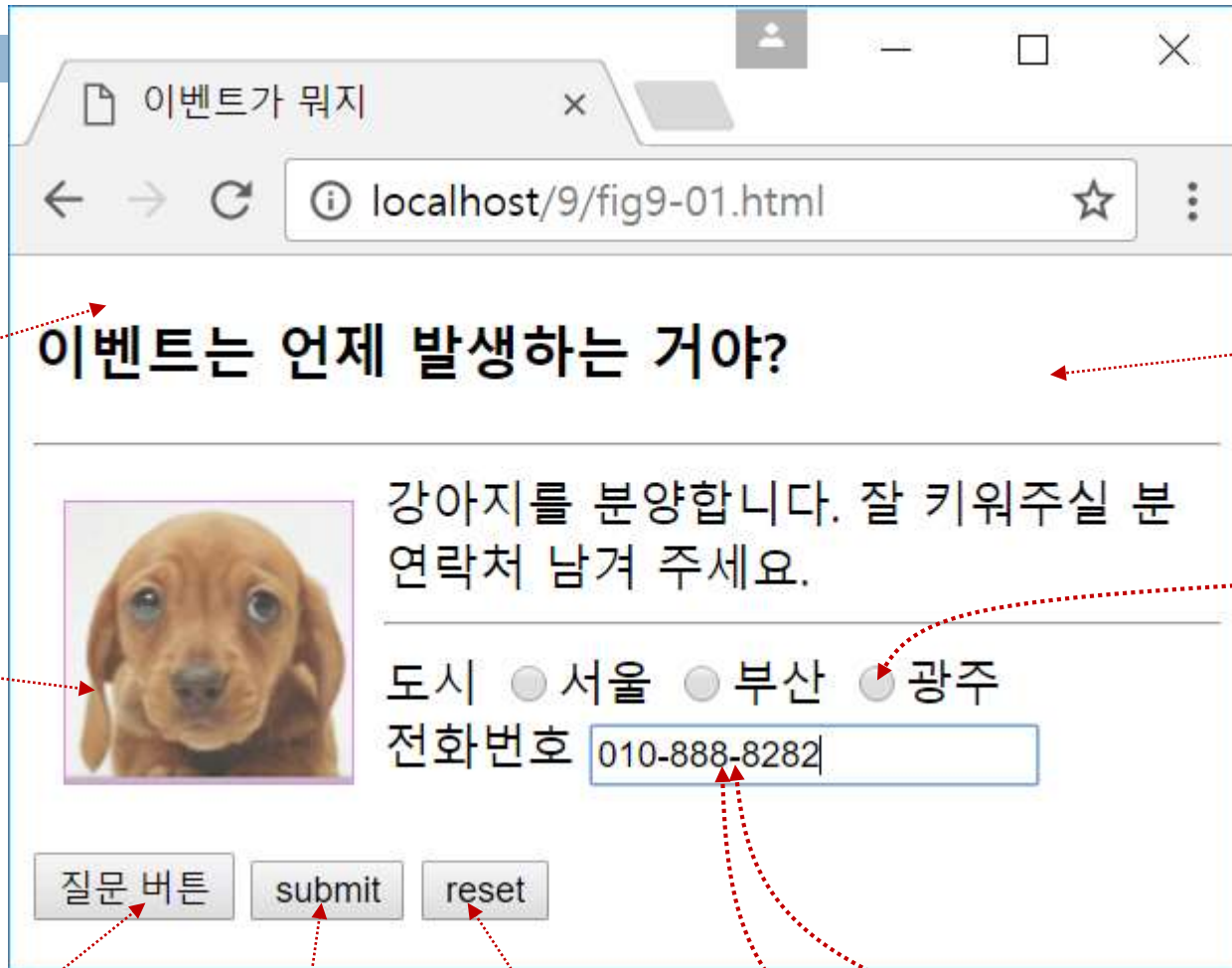
이벤트 개요

11

- 이벤트
 - ▣ 마우스 클릭, 키보드 입력, 이미지나 HTML 문서의 로딩, 타이머의 타임아웃 등 사용자의 입력 행위나 문서나 브라우저의 상태 변화를 자바스크립트 코드에게 알리는 통지(notification)
- 이벤트 리스너 (**on~~~~**)
 - ▣ 발생한 이벤트에 대처하기 위해 작성된 자바스크립트 코드
- 이벤트 종류
 - ▣ HTML5에서 이벤트 종류는 70여가지
 - ▣ **이벤트 리스너 이름은 이벤트 이름 앞에 on을 덧붙임**
 - ▣ 예) **onmousedown** : mousedown 이벤트의 리스너
onkeydown : keydown 이벤트의 리스너

브라우저에 발생하는 다양한 이벤트들

12



load 이벤트
(HTML 문서 전체 로딩 완료 시)

dblclick 이벤트
(마우스 더블클릭 시)

load 이벤트
(이미지의 로딩 완료 시)

change 이벤트
(라디오버튼 선택 시)

resize 이벤트
(윈도우 크기 변경 시)

click 이벤트
(마우스 클릭 시)

submit 이벤트
(submit 버튼 클릭 시)

reset 이벤트
(reset 버튼 클릭 시)

keypress 이벤트
(키를 누를 때)

keyup 이벤트
(누른 키를 놓을 때)

이벤트 리스너 만들기

13

□ 3 가지 방법

- HTML 태그 내에 작성 (**inline event listener**)
- DOM 객체의 **이벤트 리스너 프로퍼티**에 작성
- DOM 객체의 **addEventListener()** 메소드 이용

□ 1. HTML 태그 내에 이벤트 리스너 작성

- HTML 태그의 이벤트 리스너 속성에 리스너 코드 직접 작성
예) <p>태그에 마우스 올리면 orchid, 내리면 흰색으로 배경변경

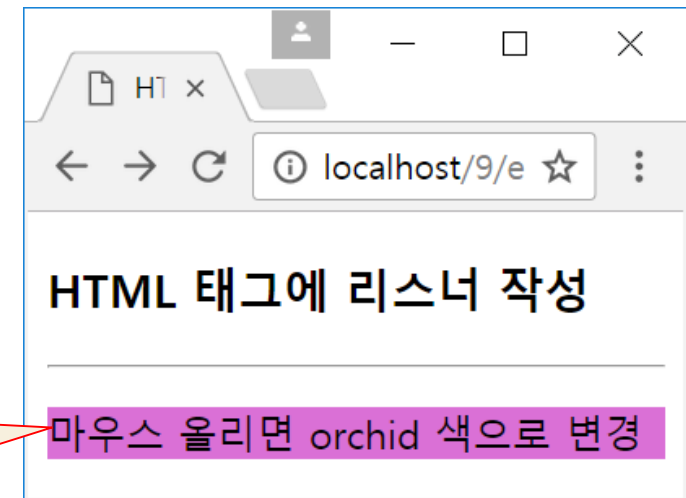
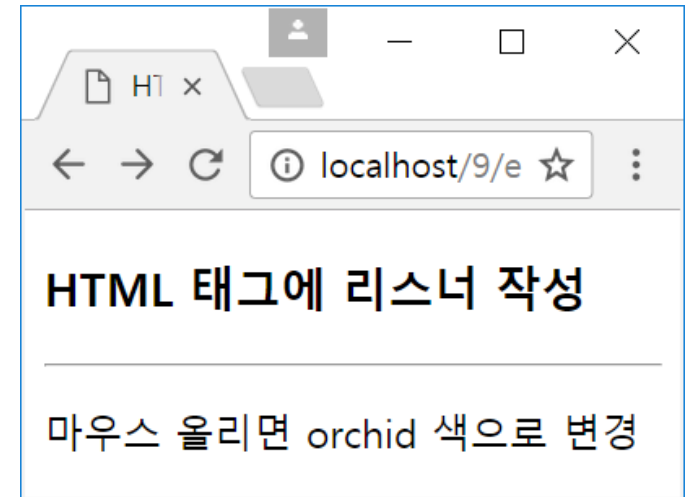
```
<p onmouseover="this.style.backgroundColor='orchid'"  
    onmouseout="this.style.backgroundColor='white'">  
    마우스 올리면 orchid 색으로 변경  
</p>
```

예제 9-1 HTML 태그 내에 이벤트 리스너 작성

14

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML 태그에 리스너 작성</title>
</head>
<body>
  <p>HTML 태그에 리스너 작성</p>
  <hr>
  <p onmouseover="this.style.backgroundColor='orchid'"
    onmouseout="this.style.backgroundColor='white'">
    마우스 올리면 orchid 색으로 변경</p>
</body>
</html>
```

wk13js_ex01_tag_event.html



이곳에 마우스를 올리면
배경색 변함

DOM 객체의 이벤트 리스너 프로퍼티에 작성

15

□ 2. DOM 객체의 이벤트 리스너 프로퍼티에 이벤트 리스너 코드 작성

예)

```
<p id="p">마우스 올리면 orchid 색으로 변경</p>
```

```
function over() { // onmouseover 리스너로 사용할 함수  
  ...  
}
```

```
var p = document.getElementById("p");  
p.onmouseover = over; // onmouseover 리스너로 over() 함수 등록
```

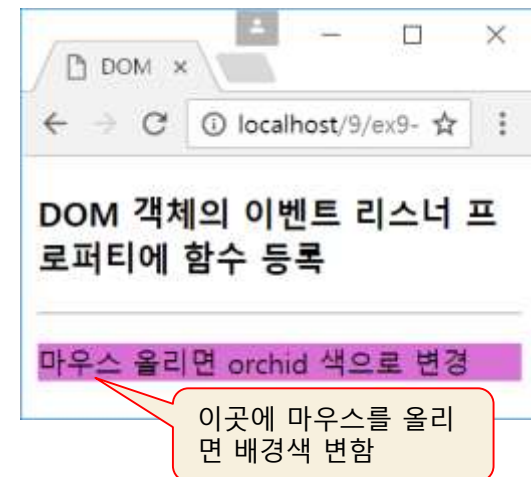
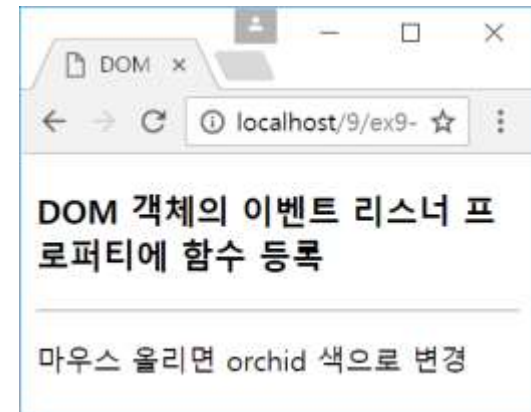
```
p.onmouseover = over(); // 잘못된 코드
```

예제 9-2 DOM 객체의 이벤트 리스너 프로퍼티에 리스너 등록

16

```
<!DOCTYPE html>
<html>
<head>
<title>DOM 객체의 이벤트 리스너 프로퍼티에 함수 등록</title>
<script>
var p;
function init() { // 문서가 완전히 로드되었을 때 호출
    p = document.getElementById("p");
    p.onmouseover = over; // over()를 onmouseover 리스너로 등록
    p.onmouseout = out; // out()를 onmouseout 리스너로 등록
}
function over() {
    p.style.backgroundColor="orchid";
}
function out() {
    p.style.backgroundColor="white";
}
</script>
</head>
<body onload="init()">
<h3>DOM 객체의 이벤트 리스너 프로퍼티에 함수 등록</h3>
<hr>
<p id="p">마우스 올리면 orchid 색으로 변경</p>
</body>
</html>
```

wk13js_ex02_dom_event.html



DOM 객체의 addEventListener() 메소드 활용

17

□ 3. addEventListener() 메소드

```
addEventListener(eventName, listener[, useCapture])
```

- eventName : 이벤트 타입을 나타내는 문자열. click, load, keydown 등
- listener : 이벤트 리스너로 등록할 함수 이름
- useCapture : true이면 이벤트 흐름 중 캡처 단계에서 실행될 리스너(listener 함수) 등록.
false이면 버블 단계에서 실행될 리스너 등록. 생략 가능하며 디폴트는 false.
이벤트 흐름은 3절에서 자세히 설명

listener 함수를 eventName의 이벤트를 처리할 리스너로 등록한다.

예)

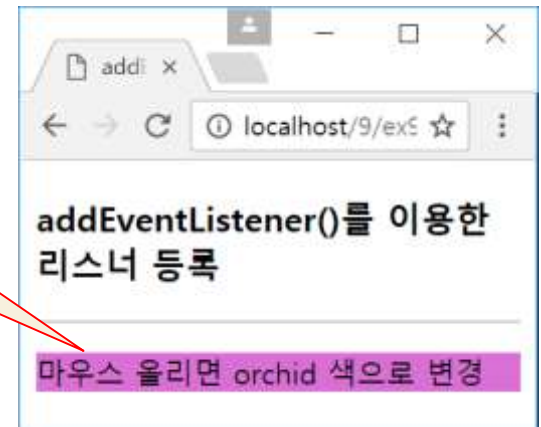
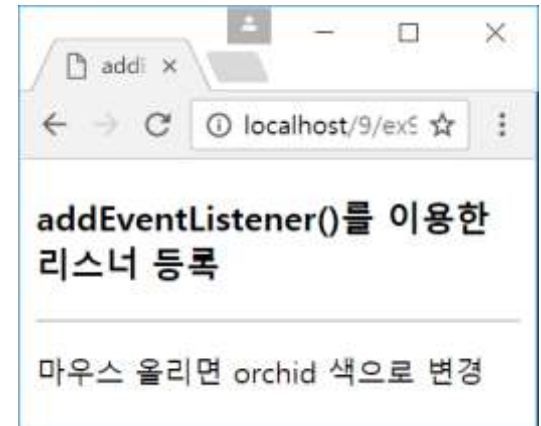
```
p.addEventListener("mouseover", over); // onmouseover 리스너로 over() 등록
```

예제 9-3 addEventListener() 사용

18

```
<!DOCTYPE html>
<html>
<head>
<title>addEventListener()를 이용한 리스너 등록</title>
<script>
var p;
function init() { // 문서가 완전히 로드되었을 때 호출
    p = document.getElementById("p");
    p.addEventListener("mouseover", over); // 이벤트 리스너 등록
    p.addEventListener("mouseout", out); // 이벤트 리스너 등록
}
function over() {
    p.style.backgroundColor="orchid";
}
function out() {
    p.style.backgroundColor="white";
}
</script>
</head>
<body onload="init()">
<h3>addEventListener()를 이용한 리스너 등록</h3>
<hr>
<p id="p">마우스 올리면 orchid 색으로 변경</p>
</body>
</html>
```

이곳에 마우스를 올리면
배경색 변함



익명 함수로 이벤트 리스너 작성

19

□ 4. 익명 함수 (anonymous function)

- ▣ 함수 이름 없이 필요한 곳에 함수의 코드를 바로 작성

예-1)

```
p.onmouseover = function () { this.style.backgroundColor = "orchid"; }; // 익명 함수
```

예-2)

```
p.addEventListener("mouseover",  
    function () { this.style.backgroundColor = "orchid"; } // 익명 함수  
);
```

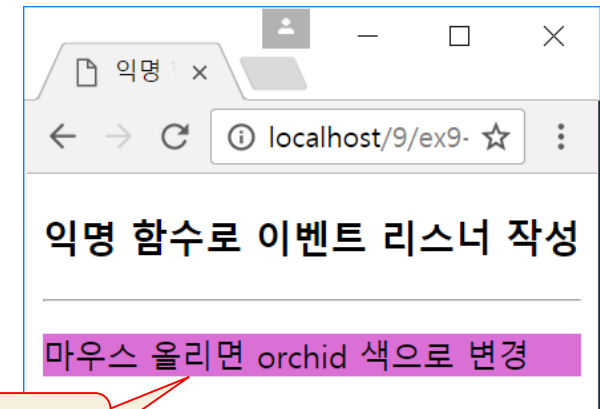
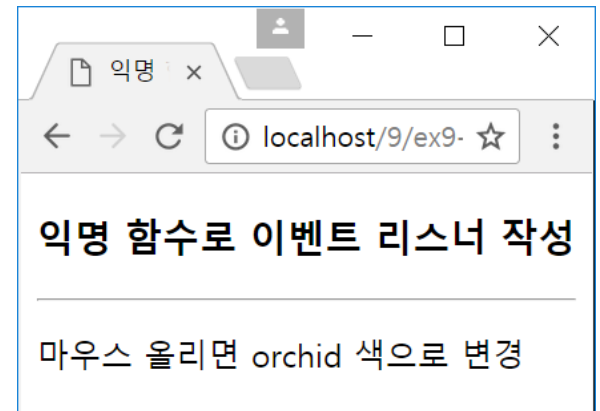
- ▣ 코드가 짧거나 한 곳에서만 사용하는 경우, 익명 함수 편리

예제 9-4 익명 함수로 이벤트 리스너 작성

20

```
<!DOCTYPE html>
<html>
<head>
<title>익명 함수로 이벤트 리스너 작성</title>
<script>
var p;
function init() { // 문서가 완전히 로드되었을 때 호출
  p = document.getElementById("p");
  p.onmouseover = function () { // 익명 함수
    this.style.backgroundColor = "orchid";
  };
  p.addEventListener("mouseout",
    function () { this.style.backgroundColor="white"; } // 익명 함수
  );
}
</script>
</head>
<body onload="init()">
<h3>익명 함수로 이벤트 리스너 작성</h3>
<hr>
<p id="p">마우스 올리면 orchid 색으로 변경</p>
</body>
</html>
```

wk13js_ex04_anonymous_listener.html



이곳에 마우스를 올리
면 배경색 변함

이벤트 리스너 작성 방법 4 가지 비교 (exam)

21

```
function over() {  
    p.style.backgroundColor="orchid";  
}
```

(1) HTML 태그
inline event

```
<p id="p" onmouseover="this.style.backgroundColor='orchid'"  
  마우스 올리면 orchid 색으로 변경  
</p>
```

(2) 이벤트 리스너
프로퍼티

```
function over() {  
    p.style.backgroundColor="orchid";  
}  
p.onmouseover = over;
```

(3) addEventListener()
메소드 이용

```
p.addEventListener("mouseover", over);
```

(4) 익명 함수 이용 I.

```
p.onmouseover = function () { this.style.backgroundColor="orchid"; };
```

(5) 익명 함수 이용 II.

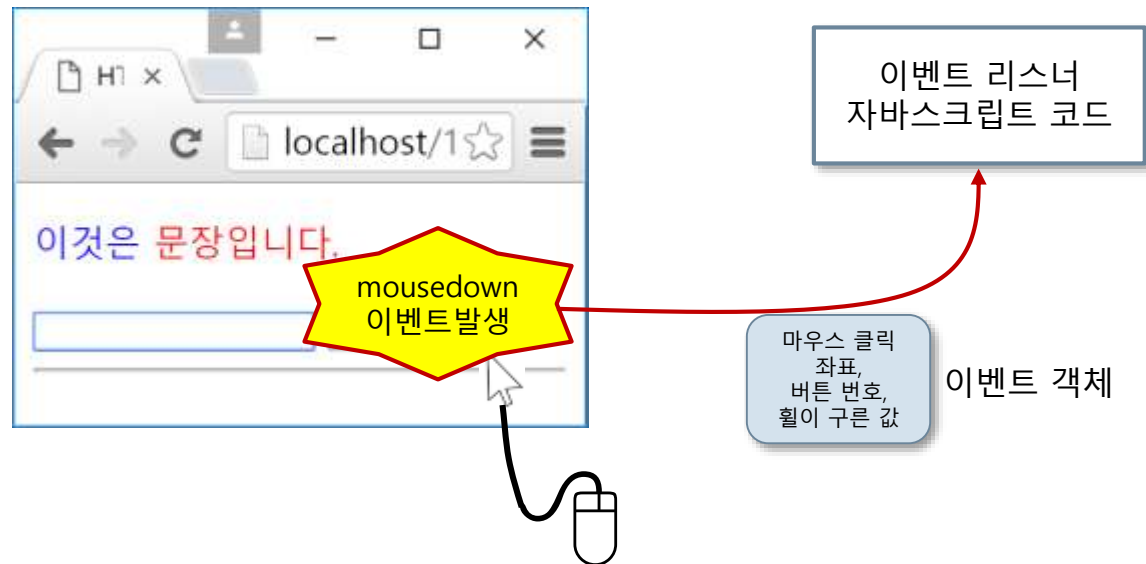
```
p.addEventListener("mouseover",  
  function () { this.style.backgroundColor="orchid"; }  
);
```

Event object

이벤트 객체

23

- 이벤트 객체 (**event object**)
 - ▣ 발생한 이벤트에 관련된 다양한 정보를 담은 객체
 - ▣ 예) mousedown 이벤트의 경우, 마우스 좌표와 버튼 번호 등
keydown 이벤트의 경우, 키 코드 값 등
 - ▣ 이벤트가 처리되고 나면 이벤트 객체 소멸



이벤트 객체 전달받기

24

- 이벤트 객체는 이벤트 리스너 함수의 첫 번째 매개변수에 전달

1. 이름을 가진 이벤트 리스너

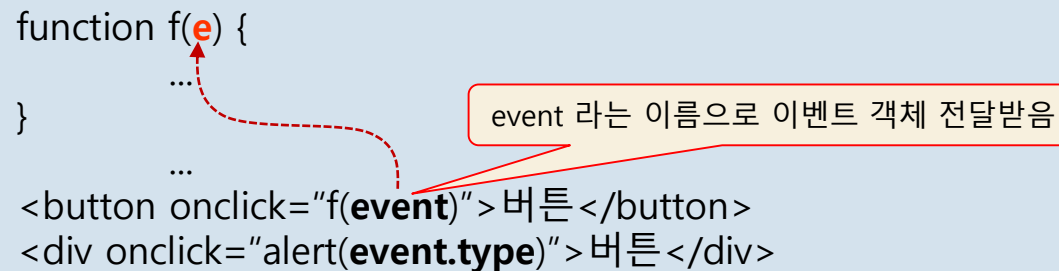
```
function f(e) { // 매개변수 e에 이벤트 객체 전달받음
    ...
}
obj.onclick = f; // obj 객체의 onclick 리스너로 함수 f 등록
```

2. 익명 함수의 경우

```
obj.onclick = function(e) { // 매개변수 e에 이벤트 객체 전달받음
    ...
}
```

3. HTML 태그에 이벤트 리스너 : **event** 라는 이름으로 전달

```
function f(e) {
    ...
}
...
<button onclick="f(event)">버튼</button>
<div onclick="alert(event.type)">버튼</div>
```



A red dashed arrow points from the **event** text in the HTML onclick attribute to the parameter **e** in the function definition. A yellow callout box with a red border contains the text "event 라는 이름으로 이벤트 객체 전달받음" (Received event object by the name event).

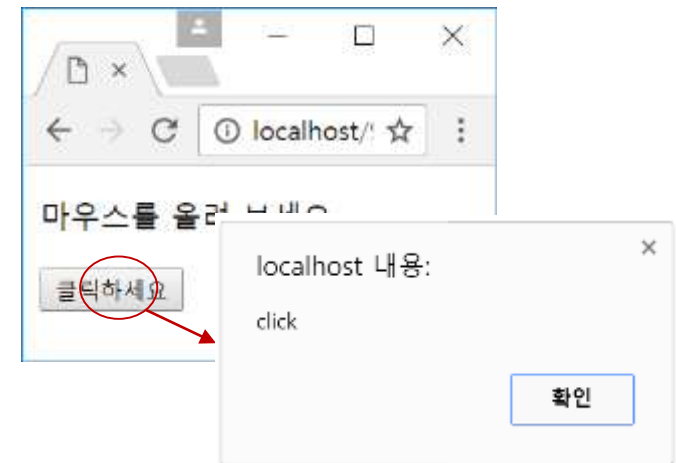
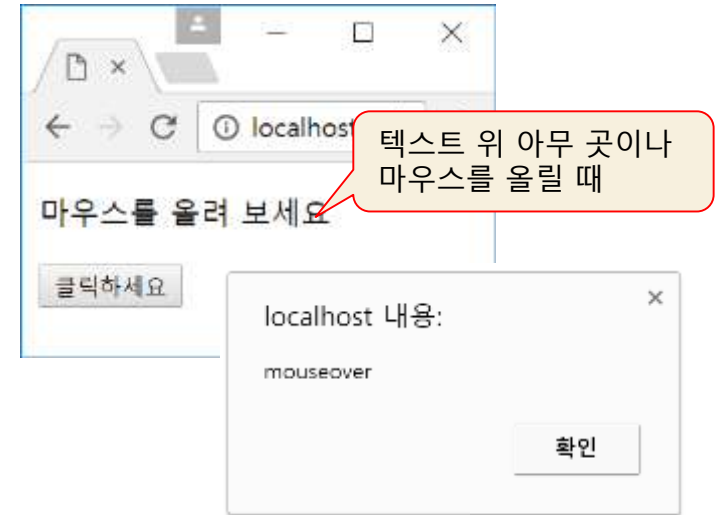
예제 9-5 이벤트 리스너에서 이벤트 객체 전달 받기

25

```
<!DOCTYPE html>
<html>
<head>
<title>이벤트 객체 전달받기</title>
</head>
<body>
<p id="p">마우스를 올려 보세요</p>
<button onclick="f(event)">클릭하세요</button>
<script>
function f(e) { // e는 현재 발생한 이벤트 객체
    alert(e.type); // 이벤트 종류 출력
}

document.getElementById("p").onmouseover = f;
</script>
</body>
</html>
```

wk13js_ex05_event_obj.html



이벤트 객체에 들어 있는 정보

26

- 이벤트 객체에 들어 있는 정보
 - ▣ 현재 발생한 이벤트에 관한 다양한 정보
 - 이벤트 객체의 프로퍼티와 메소드로 알 수 있음
 - ▣ 이벤트의 종류마다 조금씩 다름
 - 이벤트 객체의 공통 멤버

멤버	종류	설명
type	프로퍼티	현재 발생한 이벤트의 종류를 나타내는 문자열(click, load 등)
target	프로퍼티	이벤트를 발생시킨 객체(DOM 객체 혹은 HTML 태그)
currentTarget	프로퍼티	현재 이벤트 리스너를 실행하고 있는 DOM 객체
defaultPrevented	프로퍼티	이벤트의 디폴트 행동이 취소되었는지를 나타내는 true/false 값
preventDefault()	메소드	이벤트의 디폴트 행동을 취소시키는 메소드

- ▣ target 프로퍼티
 - 이벤트 타겟 객체 가리킴
 - 이벤트 타겟 : 이벤트를 유발시킨 DOM 객체
 - `<button>` 태그의 버튼을 클릭하였으면, 이때 click 이벤트의 이벤트 타겟은 버튼

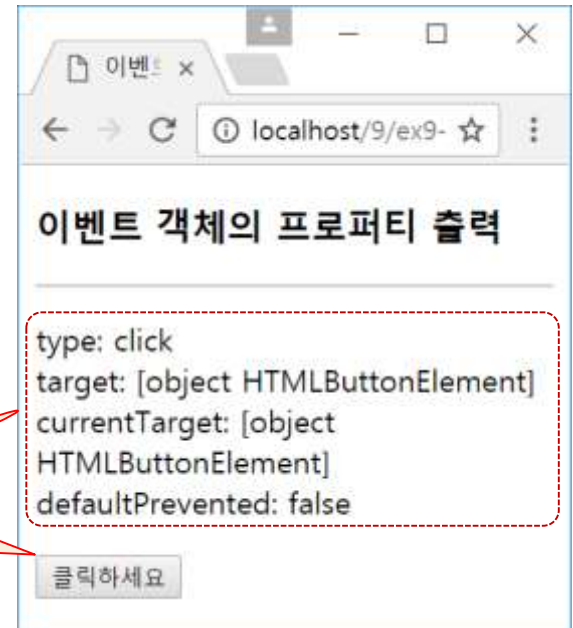
예제 9-6 이벤트 객체의 프로퍼티 출력

27

```
<!DOCTYPE html>
<html>
<head>
<title>이벤트 객체 프로퍼티</title>
</head>
<body>
<h3>이벤트 객체의 프로퍼티 출력</h3>
<hr>
<p id="p">버튼을 클릭하면 이벤트 객체를 출력합니다.</p>
<button onclick="f(event)">클릭하세요</button>
<script>
function f(e) { // e는 현재 발생한 이벤트 객체
    var text = "type: " + e.type + "<br>"
        + "target: " + e.target + "<br>"
        + "currentTarget: " + e.currentTarget + "<br>"
        + "defaultPrevented: " + e.defaultPrevented;

    var p = document.getElementById("p");
    p.innerHTML = text; // 이벤트 객체의 프로퍼티 출력
}
</script>
</body>
</html>
```

wk13js_ex06_event_obj_property.html



이벤트의 디폴트 행동 취소, preventDefault()

28

- 이벤트의 디폴트 행동이란?
 - 특정 이벤트에 대한 HTML 태그의 기본 행동
 - 사례
 - <a>의 click 이벤트의 디폴트 행동 : 웹 페이지 이동
 - Submit 버튼의 click 이벤트의 디폴트 행동 : 폼 데이터 전송
 - <input type="checkbox">의 click 이벤트의 디폴트 행동 : 체크박스선택

- 이벤트의 디폴트 행동을 막는 방법

- 1. 이벤트 리스너에서 **false** 리턴

```
<a href="http://www.naver.com" onclick="return false">  
    이동 안되는 링크  
</a>
```

- 2. 이벤트 객체의 preventDefault() 메소드 호출

```
<a href="http://www.naver.com" onclick="event.preventDefault();">  
    이동 안되는 링크  
</a>
```

- 이벤트 객체의 cancelable 프로퍼티가 true인 경우만 취소 가능

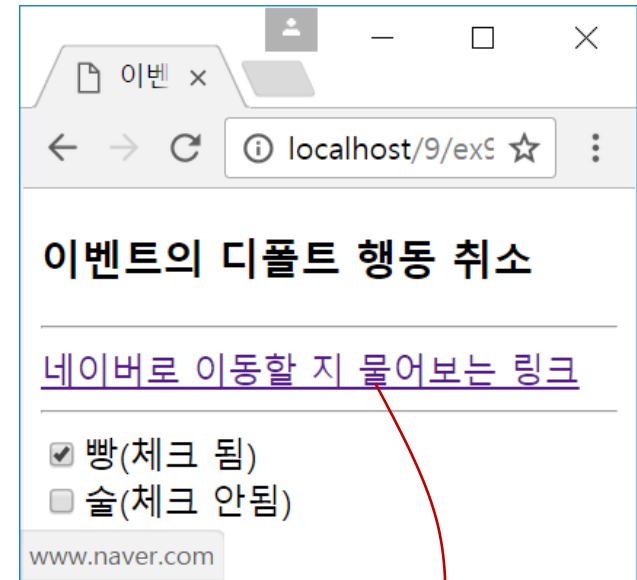
예제 9-7 이벤트의 디폴트 행동 취소

29

```
<!DOCTYPE html>
<html><head><title>이벤트의 디폴트 행동 취소</title>
<script>
function query() {
    var ret = confirm("네이버로 이동하시겠습니까?");
    return ret; // confirm()의 리턴 값은 true 또는 false
}

function noAction(e) {
    e.preventDefault(); // 이벤트의 디폴트 행동 강제취소
}
</script>
</head>
<body>
<h3>이벤트의 디폴트 행동 취소</h3>
<hr>
<a href="http://www.naver.com"
    onclick="return query()">
    네이버로 이동할 지 물어보는 링크 </a>
<hr>
<form>
    <input type="checkbox">빵(체크 됨)<br>
    <input type="checkbox"
        onclick="noAction(event)">술(체크 안됨)
</form>
</body>
</html>
```

wk13js_ex07_preventDefault.html



취소 버튼을 누르면
네이버로 이동하지 않음

Event flow

이벤트 흐름

31

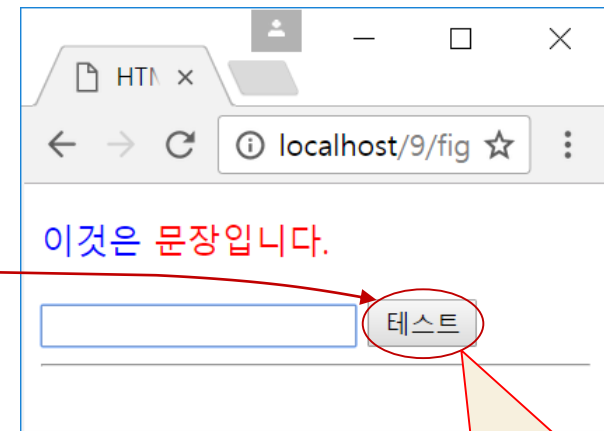
- 이벤트 흐름이란?
 - 이벤트가 발생하면 window 객체에 먼저 도달하고, DOM 트리를 따라 이벤트 타겟에 도착하고, 다시 반대 방향으로 흘러 window 객체에 도달한 다음 사라지는 과정
- 이벤트가 흘러가는 과정
 - 캡처 단계(**capturing** phase)
 - 이벤트가 window 객체에서 중간의 모든 DOM 객체를 거쳐 타겟 객체에 전달되는 과정 → 이벤트 발생 준비 단계
 - 이벤트가 거쳐가는 모든 DOM 객체(window포함)의 이벤트 리스너 실행
 - 버블 단계(**bubbling** phase)
 - 이벤트가 타겟에서 중간의 모든 DOM 객체를 거쳐 window 객체에 전달되는 과정 → 이벤트 처리 단계
 - 이벤트가 거쳐가는 모든 DOM 객체(window포함)의 이벤트 리스너 실행
- DOM 객체에는 캡처 리스너와 버블 리스너 두 개 모두 작성할 수 있음

이벤트 흐름 사례

32

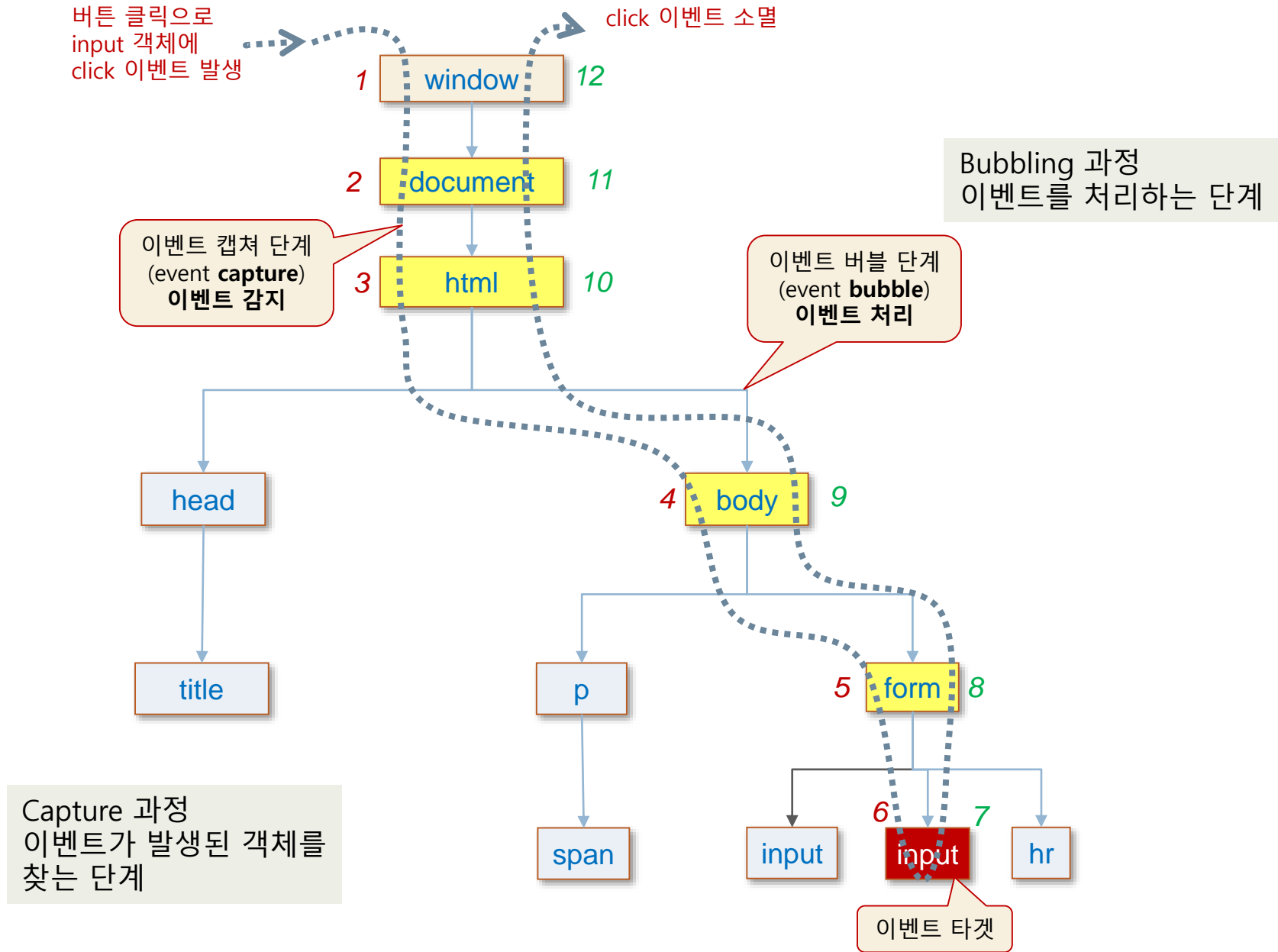
□ 샘플 웹 페이지

```
<!DOCTYPE html>
<html> <head> <title>HTML DOM 트리</title> </head>
<body>
<p style="color:blue" >이것은
  <span style="color:red">문장입니다.</span>
</p>
<form>
  <input type="text">
  <input type="button" value="테스트" id="button">
  <hr>
</form>
</body> </html>
```



버튼 클릭, click 이벤트 발생

이벤트 감지(capture) 과정과 처리(bubbling) 과정



캡처 리스너와 버블 리스너 (skip)

34

- DOM 객체의 이벤트 리스너
 - ▣ 캡처 리스너와 버블 리스너를 모두 소유 가능
 - 이벤트 리스너 등록 시, 캡처 리스너인지 버블 리스너인지 구분
- 캡처 리스너와 버블 리스너 등록
 - ▣ `addEventListener()`의 3 번째 매개 변수 이용
 - `true`이면 캡처 리스너, `false`이면 버블 리스너

```
var b = document.getElementById("button");  
b.addEventListener("click", capFunc, true); // 캡처 단계에서 capFunc() 실행  
b.addEventListener("click", bubbleFunc, false); // 버블 단계에서 bubbleFunc() 실행
```

- ▣ 다른 방법의 이벤트 리스너 등록의 경우
 - 버블 리스너로 자동 등록
 - 예)

```
obj.onclick = function(e) { // 버블 리스너도 작동  
    ...  
}
```

예제 9-8 이벤트 흐름 (skip)

35

```
<!DOCTYPE html>
<html><head><title>이벤트 흐름</title></head>
<body>
<p style="color:blue">이것은
  <span style="color:red" id="span">문장입니다.
</span>
</p>
<form>
  <input type="text" name="s">
  <input type="button" value="테스트" id="button">
</form>
<div id="div" style="color:green"></div>
<script>
var div = document.getElementById("div"); // 이벤트 메시지 출력 공간
var button = document.getElementById("button");

// body 객체에 캡처 리스너 등록
document.body.addEventListener("click", capture, true); // 캡처 단계(1)

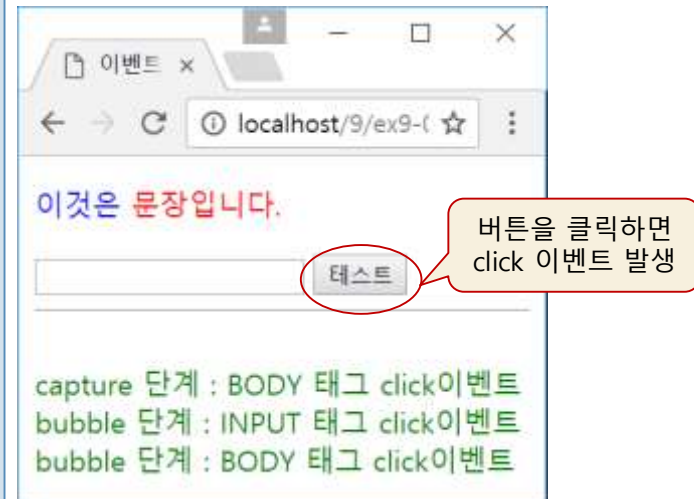
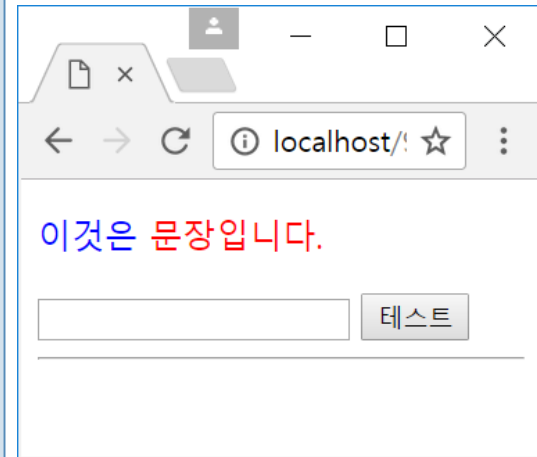
// 타겟 객체에 버블 리스너 등록
button.addEventListener("click", bubble, false); // 버블 단계(2)

// body 객체에 버블 리스너 등록
document.body.addEventListener("click", bubble, false); // 버블 단계(3)

function capture(e) { // e는 이벤트 객체
  var obj = e.currentTarget; // 현재 이벤트를 받은 DOM 객체
  var tagName = obj.tagName; // 태그 이름
  div.innerHTML += "<br>capture 단계 : " + tagName + " 태그 " + e.type + "이벤트";
}

function bubble(e) { // e는 이벤트 객체
  var obj = e.currentTarget; // 현재 이벤트를 받은 DOM 객체
  var tagName = obj.tagName; // 태그 이름
  div.innerHTML += "<br>bubble 단계 : " + tagName + " 태그 " + e.type + "이벤트";
}
</script>
</body></html>
```

wk13js_ex08_event_flow.html



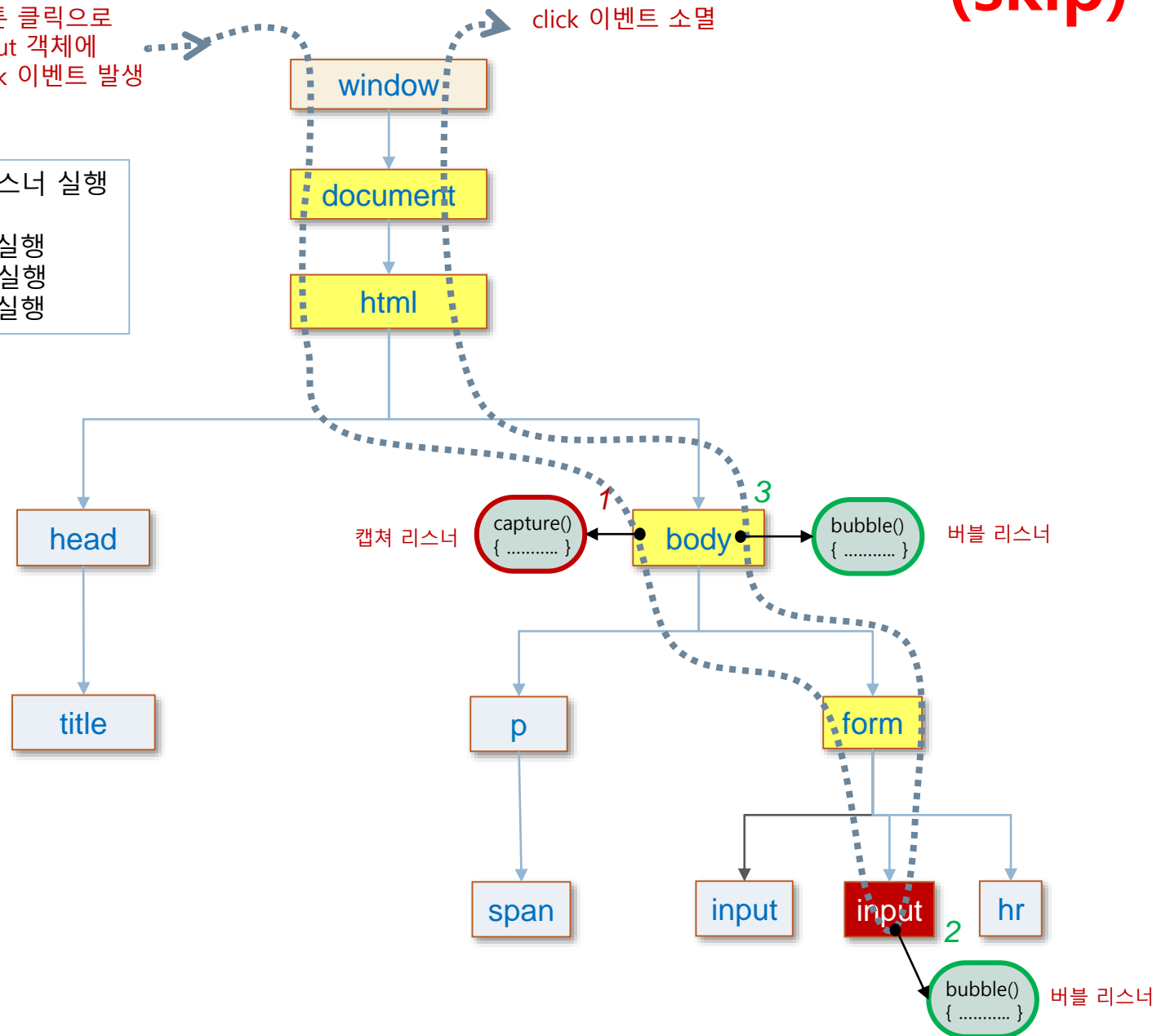
(skip)

버튼 클릭으로
input 객체에
click 이벤트 발생

click 이벤트 소멸

예제 9-8 웹 페이지의 이벤트 리스너 실행

1. <body> 태그의 캡처 리스너 실행
2. <input> 태그의 버블 리스너 실행
3. <body> 태그의 버블 리스너 실행



이벤트 흐름을 중단시킬 수 있는가? YES (skip)

37

- 이벤트 객체의 stopPropagation() 호출
 - ▣ event.stopPropagation(); // event가 이벤트 객체일 때

Mouse event

마우스 핸들링

39

□ 마우스 이벤트 객체의 프로퍼티

프로퍼티	
x, y	(x, y)는 타겟 객체의 부모 객체 내에서의 마우스 좌표
clientX, clientY	(clientX, clientY)는 브라우저 윈도우의 문서출력 영역 내에서의 마우스의 좌표
screenX, screenY	(screenX, screenY)는 스크린을 기준으로 한 마우스 좌표
offsetX, offsetY	(offsetX, offsetY)는 타겟 객체 내에서의 마우스 좌표
button	눌려진 마우스 버튼 • 0 : 아무 버튼도 눌러지지 않았음 • 1 : 왼쪽 버튼이 눌러졌음 • 2 : 오른쪽 버튼이 눌러졌음 • 3 : 왼쪽, 오른쪽 버튼이 모두 눌러졌음 • 4 : 중간 버튼이 눌러졌음
wheelDelta	마우스 휠이 구른 방향 • 양수 : 위쪽으로 굴린 경우(실제 wheelDelta 값은 120) • 음수 : 아래쪽으로 굴린 경우(실제 wheelDelta 값은 -120)

□ onclick

- HTML 태그가 클릭될 때

□ ondblclick

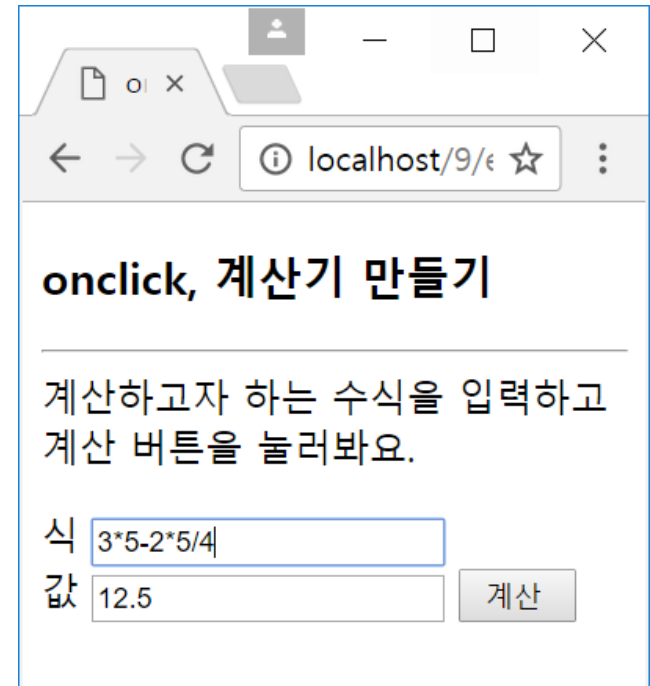
- HTML 태그가 더블클릭될 때

예제 9-9 onclick 리스너로 계산기 만들기

40

```
<!DOCTYPE html>
<html> <head>
<title>onclick</title>
<script>
function calculate() {
  var exp = document.getElementById("exp");
  var result = document.getElementById("result");
  result.value = eval(exp.value);
}
</script>
</head>
<body >
<h3> onclick, 계산기 만들기</h3>
<hr>
계산하고자 하는 수식을
입력하고 계산 버튼을 눌러봐요!
<p>
<form>
식 <input type="text" id="exp" value=""> <br>
값 <input type="text" id="result">
<input type="button" value=" 계산 "
  onclick="calculate()">
</form>
</body>
</html>
```

wk13js_ex09_event_calc.html



여러 마우스 관련 이벤트 리스너

41

▣ 마우스 관련 이벤트 리스너 호출 경우

- onmousedown : 마우스 버튼을 누르는 순간
- onmouseup : 눌려진 버튼이 놓여지는 순간
- onmouseover : 마우스가 태그 위로 올라오는 순간. 자식 영역 포함
- onmouseout : 마우스가 태그를 벗어나는 순간. 자식 영역 포함
- Onmouseenter : 마우스가 태그 위로 올라오는 순간. 버블 단계 없음
- Onmouseleave : 마우스가 태그를 떠나는 순간. 버블 단계 없음
- onwheel : HTML 태그에 마우스 휠이 구르는 동안 계속 호출
 - 위쪽으로 굴린 경우 : *wheelDelta* 프로퍼티 값 양수(120)
 - 아래쪽으로 굴린 경우 : *wheelDelta* 프로퍼티 값 음수(-120)

```
obj.onwheel = function (e) {  
    if(e.wheelDelta < 0) { // 아래쪽으로 휠을 굴린 경우  
        ...  
    }  
    else { // 위쪽으로 휠을 굴린 경우  
        ...  
    }  
};
```

예제 9-10 마우스 관련 이벤트 리스너

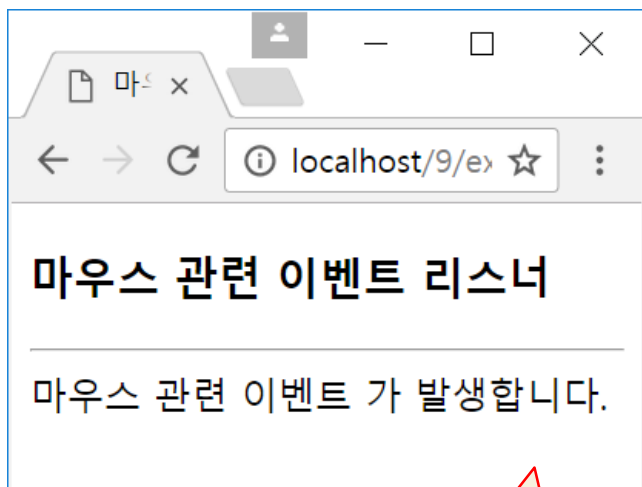
42

```
<!DOCTYPE html>
<html><head>
<title>마우스 관련 리스너</title>
<script>
var width=1; // 테두리 두께
function down(obj) {
    obj.style.fontStyle = "italic";
}
function up(obj) {
    obj.style.fontStyle = "normal";
}
function over(obj) {
    obj.style.borderColor = "violet";
    // 테두리 폭이 0일 때 색은 보이지 않는다.
}
function out(obj) {
    obj.style.borderColor = "lightgray";
}
function wheel(e, obj) { // e는 이벤트 객체
    if(e.wheelDelta < 0) { // 휠을 아래로 굴릴 때
        width--; // 폭 1 감소
        if(width < 0) width = 0; // 폭이 0보다 작아지지 않게
    }
    else // 휠을 위로 굴릴 때
        width++; // 폭 1 증가
    obj.style.borderStyle = "ridge";
    obj.style.borderWidth = width+"px";
}
</script> </head>
```

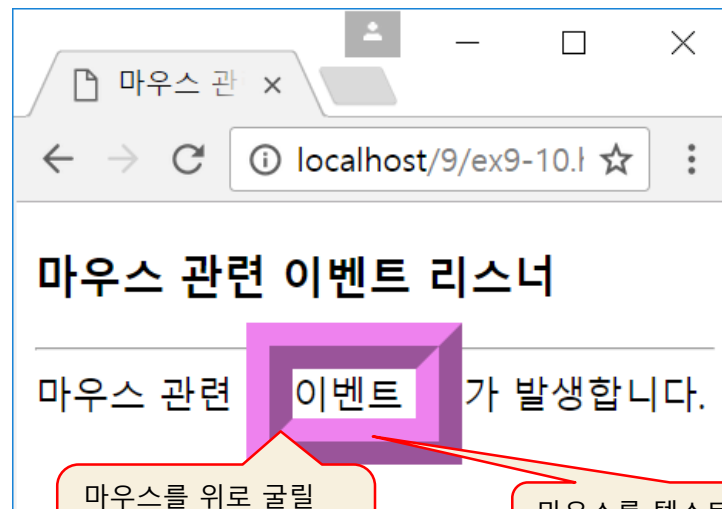
```
<body >
<h3>마우스 관련 이벤트 리스너</h3>
<hr>
<div>마우스 관련
    <span onmousedown="down(this)"
        onmouseup="up(this)"
        onmouseover="over(this)"
        onmouseout="out(this)"
        onwheel="wheel(event, this)"
        style="display:inline-block">이벤트
    </span>가 발생합니다.
</div>
</body>
</html>
```

wk13js_ex10_mouse_event.html



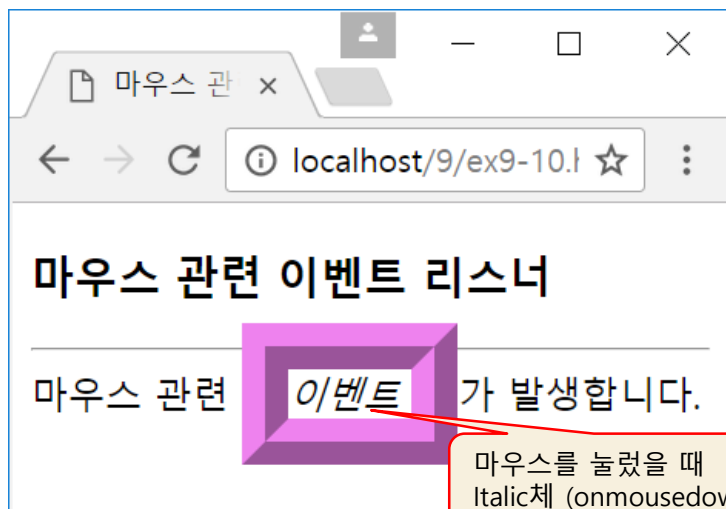


초기 화면

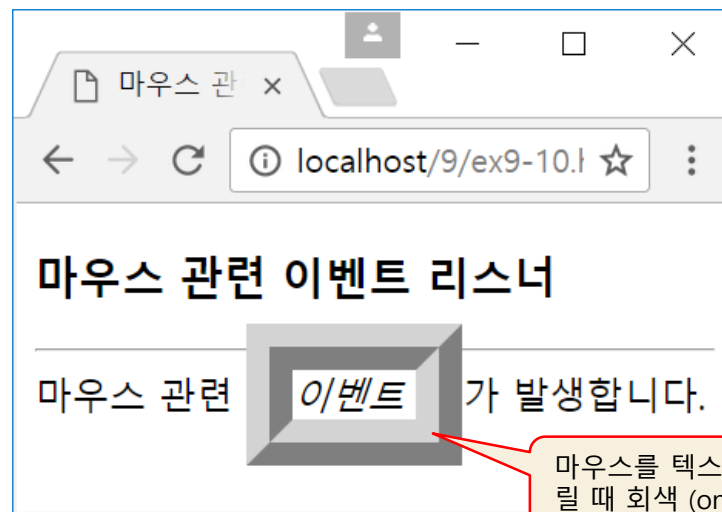


마우스를 위로 굴릴 때 두께가 1씩 두꺼워진다. (onwheel)

마우스를 텍스트위로 올릴 때 violet 색(onmouseover)



마우스를 눌렀을 때 Italic체 (onmousedown)



마우스를 텍스트에서 내릴 때 회색 (onmouseout)

예제 9-11 onmousemove와 마우스 위치 및 버튼

44

```
<!DOCTYPE html>
<html><head>
<title>마우스 이벤트 객체의 프로퍼티</title>
<style>
div {
  background : skyblue;
  width : 250px;
}
</style>
</head>
<body>
<h3>마우스 이벤트 객체의 프로퍼티와 onmousemove</h3>
<hr>
이미지 위에 마우스를 움직일 때
onmousemove 리스너가 실행되고,
마우스의 위치를 보여줍니다.<p>
<p>
<div id="div"></div>
<script>
var div = document.getElementById("div");
function where(e) {
  var text = "버튼=" + e.button + "<br>";
  text += "(screenX, screenY)=" +
    e.screenX + "," + e.screenY + "<br>";
  text += "(clientX, clientY)=" +
    e.clientX + "," + e.clientY + "<br>";
  text += "(offsetX, offsetY)=" +
    e.offsetX + "," + e.offsetY + "<br>";
  text += "(x, y)=" + e.x + "," + e.y + "Wn";
  div.innerHTML = text;
}
</script>
</body>
</html>
```






wk13js_ex11_mouse_position.html



예제 9-11 onmousemove와 마우스 위치 및 버튼

45

The numbers in the table specify the first browser version that fully supports the property.

Property					
button	Yes	Yes	Yes	Yes	8.0

Syntax

`event.button`

button property

Technical Details

Return Value:

A Number, representing which mouse button that was pressed when the mouse event occurred.

Possible values:

- 0 : Left mouse button
- 1 : Wheel button or middle button (if present)
- 2 : Right mouse button

Note: Internet Explorer 8 and earlier has different return values:

- 1 : Left mouse button
- 2 : Right mouse button
- 4 : Wheel button or middle button (if present)

Note: For a left-hand configured mouse, the return values are reversed

DOM Version:

DOM Level 2 Events

oncontextmenu

46

- HTML 태그 위에 마우스 오른쪽 버튼 클릭
 - ▣ 디폴트로 컨텍스트 메뉴(context menu) 출력
 - ‘소스 보기’나 ‘이미지 다운로드’ 등의 메뉴
 - ▣ oncontextmenu 리스너가 먼저 호출
 - **false**를 리턴하면 컨텍스트 메뉴를 출력하는 디폴트 행동 취소

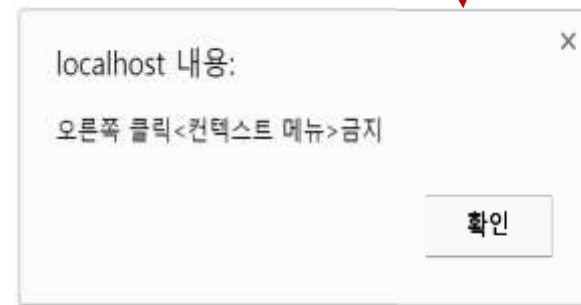
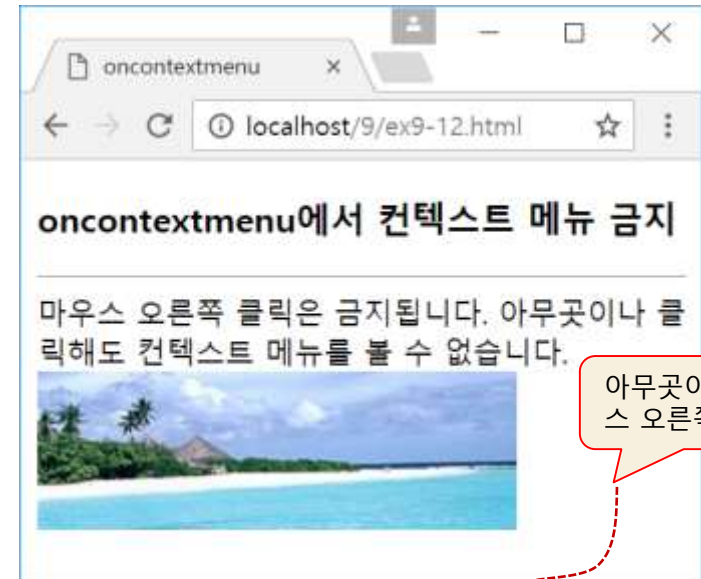
```
document.oncontextmenu = function () {  
    ...  
    return false; // 컨텍스트 메뉴 출력 금지  
}
```

예제 9-12 oncontextmenu로 소스 보기나 이미지 다운로드 금지

47

```
<!DOCTYPE html>
<html>
<head>
<title>oncontextmenu</title>
<script>
function hideMenu() {
    alert("오른쪽 클릭<컨텍스트 메뉴>금지");
    return false;
}
document.oncontextmenu=hideMenu;
</script>
</head>
<body>
<h3>oncontextmenu에서 컨텍스트 메뉴 금지</h3>
<hr>
마우스 오른쪽 클릭은 금지됩니다. 아무곳이나
클릭해도 컨텍스트 메뉴를 볼 수 없습니다.
</body>
</html>
```

wk13js_ex12_contextmenu.html



→ Check the contextmenu of an image
wk13js_ex12_contextmenu_prj.html

onload
preloading

문서의 로딩 완료와 onload

49

□ onload

- ▣ window 객체에 발생
 - 웹 페이지의 로딩 완료시 호출되는 이벤트 리스너
- ▣ onload 리스너 작성 방법
 1. `window.onload="alert('onload');";`
 2. `<body onload="alert('onload');">`

이 둘은 같은 표현임.

`<body>`에 onload를 적용한 window 객체에 load 이벤트가 전달됨

* `document.onload`는 최근에 와서 많은 브라우저에서 작동하지 않음

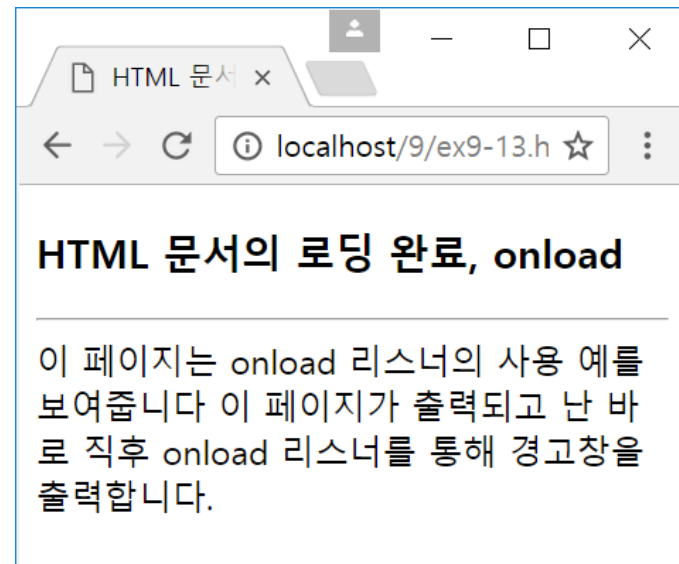
예제 9-13 onload에서 사이트 이전을 알리는 공지창 출력

50

```
<!DOCTYPE html>
<html>
<head>
<title>HTML 문서의 onload</title>
</head>
<body onload="alert('이 사이트는 2017년 9월1일부터 www.js.co.kr로 옮겨지게 됩니다.')">
<h3>HTML 문서의 로딩 완료, onload</h3>
<hr>
이 페이지는 onload 리스너의
사용 예를 보여줍니다
이 페이지가 출력되고 난 바로 직후
onload 리스너를 통해
경고창을 출력합니다.
</body>
</html>
```

₩는 뒤에 <enter>
키를 무시하게 만들

wk13js_ex13_onload.html



localhost 내용:

이 사이트는 2017년 9월1일부터 www.js.co.kr로 옮겨지게 됩니다.

확인

이미지 로딩 완료와 onload

51

□ Image 객체

- ▣ 태그에 의해 생성되는 DOM 객체
- ▣ new Image(); 자바스크립트 코드에 의해 생성되는 객체

□ onload


- ▣ 이미지의 로딩이 완료되면 Image 객체에 발생하는 이벤트

□ 새로운 이미지를 로딩하는 방법

```

```

```
var myImg = document.getElementById("myImg");  
myImg.src = "banana.png";
```

 banana.png 이미지의 로딩이 완료된 myImg의 onload 리스너 실행

이미지 로딩시 주의할 점 (check **preloading**)

52

□ 잘못된 이미지 로딩 코드

- ▣ 이미지를 로딩하여 이미지 폭을 알아내는 코드

```
var myImg = document.getElementById("myImg");  
myImg.src = "banana.png";  
var width = myImg.width;           // banana.png 이미지의 폭
```

- ▣ **문제점 (코드 실행 속도 >> 이미지 로딩 속도)**

- myImg.src = "banana.png"; **실행 직후 이미지 로딩 완료되지 않음**
- var width = myImg.width; **이미지 로딩 완료전이면, myImg.width=0**

□ 코드 수정

- ▣ onload 리스너에서 이미지 폭을 알아내는 코드 작성

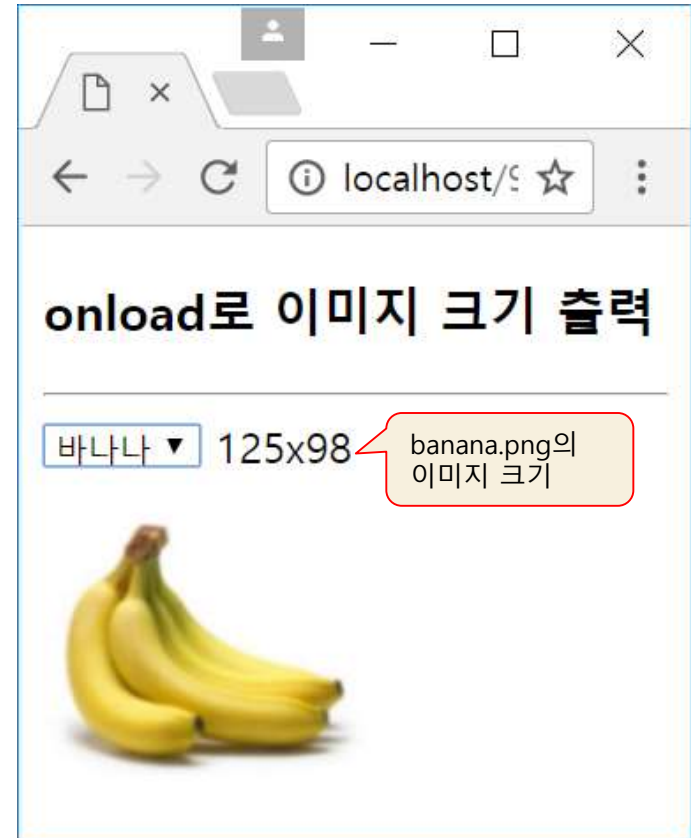
```
var myImg = document.getElementById("myImg");  
myImg.onload = function () {           // 이미지 로딩 완료 시 실행  
    var width = myImg.width;           // 정확한 이미지 폭 읽기  
}  
myImg.src = "banana.png";              // 이미지 로딩 지시
```

예제 9-14 onload로 이미지의 크기 알아내기 (exam.)

53

```
<!DOCTYPE html>
<html>
<head>
<title>onload로 이미지 크기 출력</title>
<script>
function changelImage() {
    var sel = document.getElementById("sel");
    var img = document.getElementById("myImg");
    img.onload = function () { // 이미지 크기 출력
        var mySpan = document.getElementById("mySpan");
        mySpan.innerHTML = img.width + "x" + img.height;
    }
    var index= sel.selectedIndex; // 선택된 옵션 인덱스
    img.src = sel.options[index].value; // <option>의 value 속성
}
</script>
</head>
<body onload="changelImage()">
<h3>onload로 이미지 크기 출력</h3>
<hr>
<form>
<select id="sel" onchange="changelImage()">
    <option value="media/apple.png">사과
    <option value="media/banana.png">바나나
    <option value="media/mango.png">망고
</select>
<span id="mySpan">이미지 크기</span>
</form>
<p></p>
</body>
</html>
```

wk13js_ex14_image_load.html



new Image()로 이미지 로딩과 출력

54

□ 동적으로 이미지 객체 생성

▣ new Image()

- 이미지 객체가 생겼지만 화면에 출력되지 않음

□ new Image()의 이미지 객체에 이미지 로딩

```
var bananalmg = new Image();    // 이미지 객체 생성  
bananalmg.src = "banana.png";  // 이미지 로딩
```

□ 로딩된 이미지 출력

▣ 태그에 할당된 브라우저 공간에 이미지 출력

```

```

```
var mylmg = document.getElementById("mylmg");  
mylmg.src = bananalmg.src;    // 이미지 출력
```

예제 9-15 new Image()로 이미지 프리로딩 [DIY]

55

```
<!DOCTYPE html>
<html><head>
<title>new Image()로 이미지 로딩</title>
<script>
// 미리 로딩해둘 이미지 이름 배열
var files = ["media/Penguins.png",
             "media/Lighthouse.png",
             "media/Chrysanthemum.png",
             "media/Desert.png",
             "media/Hydrangeas.png",
             "media/Jellyfish.png",
             "media/Koala.png",
             "media/Tulips.png"];
var imgs = new Array();
for(var i=0; i<files.length; i++) {
  imgs[i] = new Image(); // 메모리에 이미지 객체 생성
  imgs[i].src = files[i]; // 이미지 로딩
}

// 다음 이미지 출력
var next = 1;
function change(img) {
  img.src = imgs[next].src; // 이미지 변경
  next++; // 다음 이미지
  next %= imgs.length; // 개수를 넘으면 처음으로
}
</script> </head>
<body>
<h3>new Image()로 이미지 로딩</h3>
<hr>
이미지를 클릭하면 다음 이미지를 보여줍니다.<p>

</body> </html>
```

wk13js_ex15_image_preload.html



클릭하면 다음 이미지를 보여준다.

onblur

onfocus

onblur & onfocus

57

□ 포커스

- ▣ 포커스는 현재 키 입력에 대한 독점권
- ▣ 브라우저는 포커스를 가지고 있는 HTML 태그 요소에 키 공급

□ onblur

- ▣ **포커스를 잃을 때** 발생하는 이벤트 리스너
 - 예) 다른 HTML 요소를 클릭하면, 현재 HTML 요소는 포커스를 잃는다.

□ onfocus

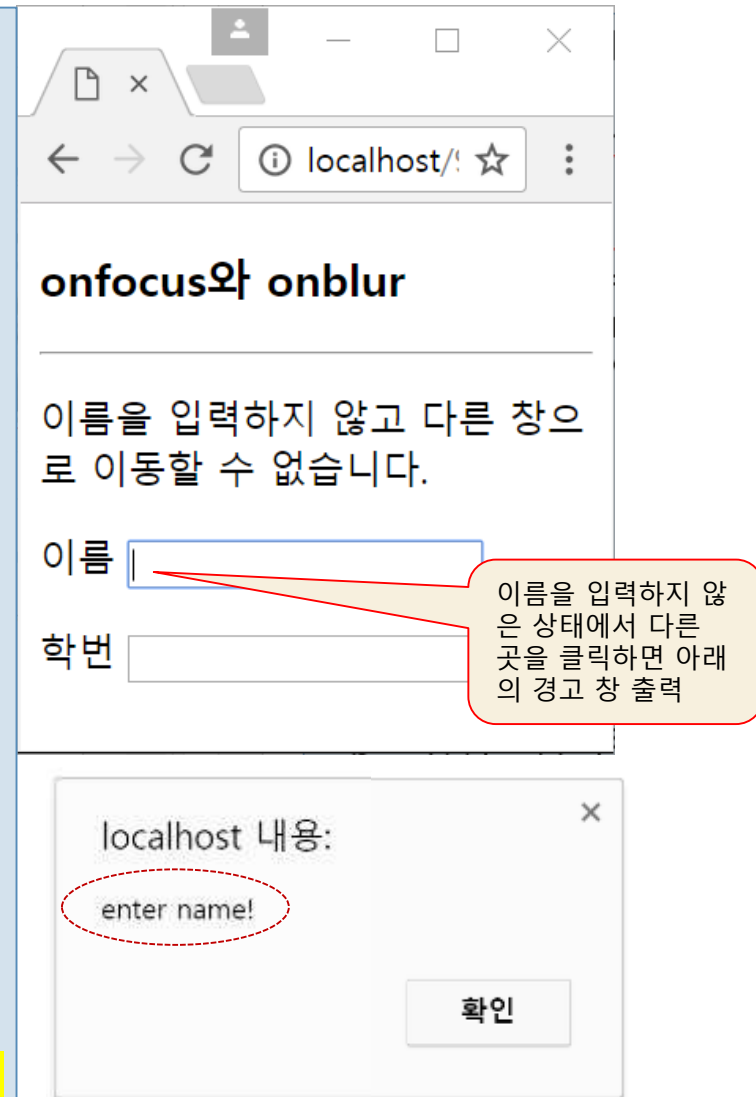
- ▣ **포커스를 얻을 때** 발생하는 이벤트 리스너
 - 예) 현재 HTML 요소를 클릭하면, 현재 HTML 요소가 포커스를 얻는다.

예제 9-16 onfocus와 onblur, 입력 없이 다른 창으로 갈 수 없음 (wrong action!)

58

```
<!DOCTYPE html>
<html>
<head>
<title>onfocus와 onblur</title>
<script>
function checkFilled(obj) {
  if(obj.value == "") {
    alert("enter name!");
    obj.focus(); // obj에 다시 포커스
  }
}
</script>
</head>
<body onload="document.getElementById('name').focus();" >
<h3>onfocus와 onblur</h3>
<hr>
<p>이름을 입력하지 않고 다른 창으로
이동할 수 없습니다.</p>
<form>
이름 <input type="text" id="name"
      onblur="checkFilled(this)"> <p>
학번 <input type="text">
</form>
</body>
</html>
```

wk13js_ex16_blur_focus.html



Event of
radio button
checkbox
select

라디오버튼과 체크박스

60

□ 라디오버튼 객체

- ▣ `<input type="radio">`로 만들어진 라디오 버튼 DOM 객체

```
<form>  
  <input type="radio" name="city" value="seoul">서울  
  <input type="radio" name="city" value="busan">부산  
  <input type="radio" name="city" value="chunchen">춘천  
</form>
```

☐ 서울 ☒ 부산 ☐ 춘천

- ▣ 라디오 버튼 객체들 알아내기

```
var kcity = document.getElementsByName("city"); // kcity[0], kcity[1], kcity[2]
```

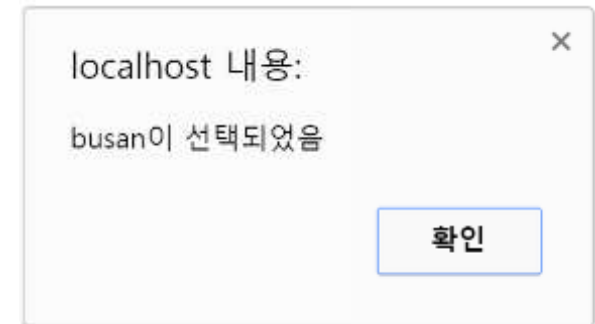
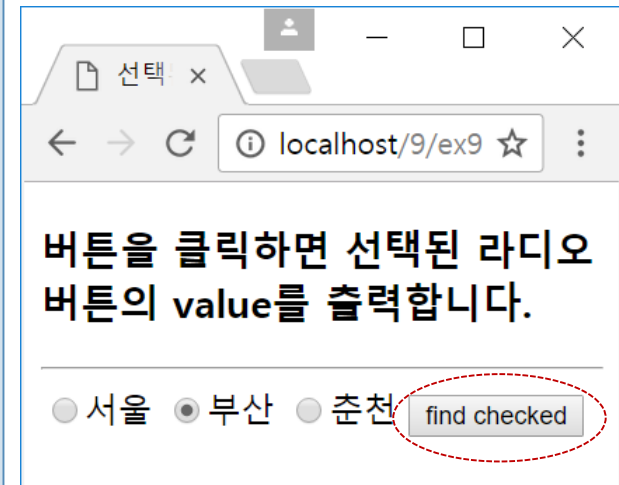
□ 체크박스 객체

- ▣ `<input type="checkbox">`로 만들어진 체크박스 DOM 객체

예제 9-17 선택된 라디오버튼 알아내기 (exam)

61

```
<!DOCTYPE html>
<html>
<head>
<title>선택된 라디오버튼 알아내기</title>
<script>
function findChecked() {
    var found = null;
    var kcity = document.getElementsByName("city");
    for(var i=0; i<kcity.length; i++) {
        if(kcity[i].checked == true)
            found = kcity[i];
    }
    if(found != null)
        alert(found.value + "이 선택되었음");
    else
        alert("선택된 것이 없음");
}
</script>
</head>
<body>
<h3>버튼을 클릭하면 선택된 라디오 버튼의 value를 출력합니다.</h3>
<hr>
<form>
    <input type="radio" name="city" value="seoul" checked>서울
    <input type="radio" name="city" value="busan">부산
    <input type="radio" name="city" value="chunchen">춘천
    <input type="button" value="find checked" onclick="findChecked()">
</form>
</body>
</html>
```

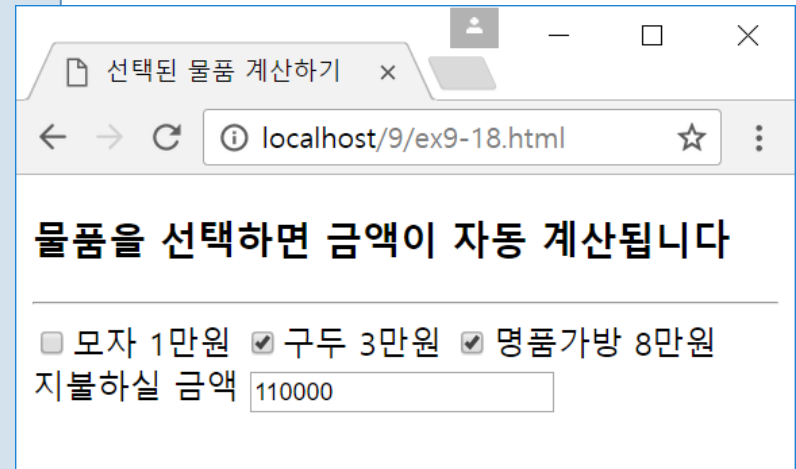


예제 9-18 체크박스로 선택한 물품 계산

62

```
<!DOCTYPE html>
<html>
<head>
<title>선택된 물품 계산하기</title>
<script>
var sum=0;
function calc(cBox) {
  if(cBox.checked)
    sum += parseInt(cBox.value);
  else
    sum -= parseInt(cBox.value);
  document.getElementById("sumtext").value = sum;
}
</script>
</head>
<body>
<h3>물품을 선택하면 금액이 자동 계산됩니다</h3>
<hr>
<form>
<input type="checkbox" name="hat" value="10000"
  onclick="calc(this)">모자 1만원
<input type="checkbox" name="shoes" value="30000"
  onclick="calc(this)">구두 3만원
<input type="checkbox" name="bag" value="80000"
  onclick="calc(this)">명품가방 8만원<br>
지불하실 금액 <input type="text" id="sumtext" value="0" >
</form>
</body>
</html>
```

wk13js_ex18_checkbox.html

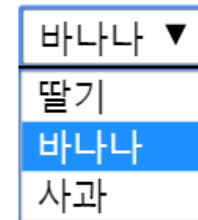


select 객체와 onchange

63

- select 객체는 <select> 태그로 만들어진 **콤보박스**
 - option 객체는 <option> 태그로 표현되는 옵션 아이템

```
<select id="fruits">  
  <option value="1">딸기</option>  
  <option value="2" selected>바나나</option>  
  <option value="3">사과</option>  
</select>
```



- 선택된 옵션 알아내기

```
var sel = document.getElementById("fruits");  
var index = sel.selectedIndex;    // index는 선택 상태의 옵션 인덱스
```

- 옵션 선택

```
sel.selectedIndex = 2;           // 3번째 옵션 "사과"의 index  
sel.options[2].selected = true;  // 3번째 옵션 "사과" 선택
```

- **select와 onchange 리스너**

- 선택된 옵션이 변경되면 select 객체의 onchange 리스너 호출

```
<select id="fruits" onchange="drawImage()">...</select>
```

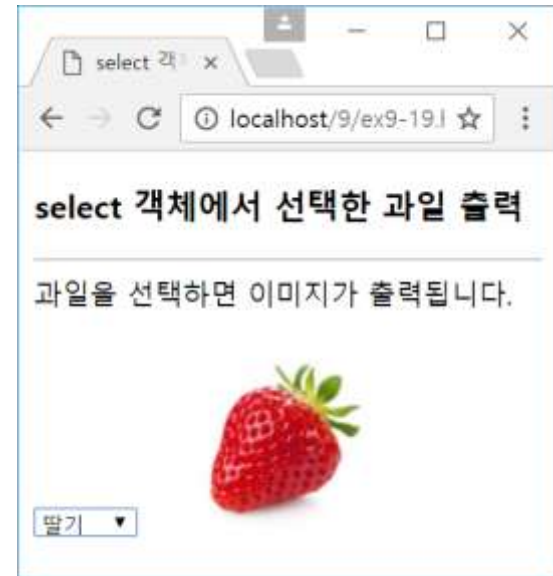

예제 9-19 select 객체에서 선택한 과일 출력 (exam)

64

```
<!DOCTYPE html>
<html>
<head>
<title>select 객체에서 선택한 과일출력</title>
<script>
function showImage() {
    var sel = document.getElementById("fruits");
    var img = document.getElementById("fruitimage");
    img.src = sel.options[sel.selectedIndex].value;
}
</script>
</head>
<body>
<h3>select 객체에서 선택한 과일 출력</h3>
<hr>
과일을 선택하면 이미지가 출력됩니다.<p>
<form>
<select id="fruits" onchange="showImage()">
    <option value="media/strawberry.png">딸기
    <option value="media/banana.png" selected>바나나
    <option value="media/apple.png">사과
</select>

</form>
</body>
</html>
```

wk13js_ex19_select.html



Try below!

<body onload="showImage()">

**key
event**

키 이벤트

66

- onkeydown, onkeypress, onkeyup
 - ▣ onkeydown
 - 키가 눌러지는 순간 호출. 모든 키에 대해 작동
 - ▣ onkeypress
 - 문자 키와 <Enter>, <Space>, <Esc> 키에 대해서만 눌러지는 순간에 추가 호출
 - 문자 키가 아닌 경우(<F1>, <Shift>, <PgDn>, , <Ins> 등) 호출되지 않음
 - ▣ onkeyup
 - 눌러진 키가 떼어지는 순간 호출

예제 9-20 키 이벤트 리스너와 이벤트 객체의 프로퍼티

67

```
<!DOCTYPE html>
<html><head>
<title>키 이벤트</title>
<script>
function whatKeyDown(e) {
    var str = "";
    var div = document.getElementById("div");
    div.innerHTML = ""; // div 객체 내용을 지운다.
    if(e.altKey) {
        if(e.altLeft) str += "왼쪽 Alt 키가 눌러졌습니다.";
        else str += "오른쪽 Alt 키가 눌러졌습니다.";
        str += "<br>";
    }

    if(e.shiftKey) {
        if(e.shiftLeft) str += "왼쪽 Shift 키가 눌러졌습니다.";
        else str += "오른쪽 Shift 키가 눌러졌습니다.";
        str += "<br>";
    }

    if(e.ctrlKey) {
        if(e.ctrlLeft) str += "왼쪽 Ctrl 키가 눌러졌습니다.";
        else str += "오른쪽 Ctrl 키가 눌러졌습니다.";
        str += "<br>";
    }

    str += String.fromCharCode(e.keyCode) + "키가 눌러졌습니다."
    div.innerHTML = str; // div 객체에 문자열을 출력한다.
}
</script>
</head>
```

```
<body>
<h3>키 리스너와 키 이벤트 객체의 프로퍼티</h3>
<hr>
텍스트 창에 키를 눌러 보세요. Alt, Shift, Ctrl 키도 가능합니다.<br>
<input type="text" id="text" onkeydown="whatKeyDown(event)">
<div id="div" style="background-color:skyblue; width:250px; height:50px">
</div>
</body>
</html>
```



DIY: hmnn_image5.html

68

[DIY] 자바스크립트 이미지 로딩에 관한 문제를 해결하는
다음 html 문서를 만드세요.

1. [예제 9-15 이미지 변경 후 정보 출력]
마음에 드는 이미지 5개를 구하여 현재 이미지를 보여주면서 이미지 관련
정보 (이미지 이름, 폭, 높이 등)를 그림 아래 출력하고, 현재 그림을 클릭하면
다음 그림과 정보를 보여주는 html 문서를 만드세요.

파일명: hmnn_image5.html

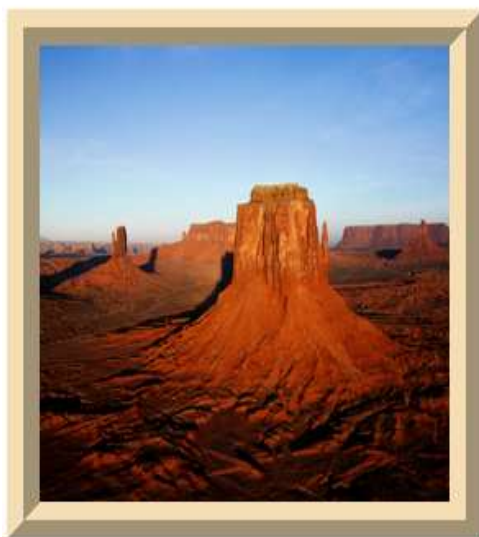
가점: Javascript 프로그래밍 응용 능력.

DIY: possible results

69

이미지 프리로딩과 정보 출력

이미지를 클릭하면 다음 이미지와 정보를 보여줍니다.



이미지 이름 : Desert.png
폭 : 512, 높이 : 384

이미지 프리로딩과 정보 출력

이미지를 클릭하면 다음 이미지와 정보를 보여줍니다.



그림 정보

Name : Penguins.png
Width : 512
Height : 384

DIY: [hmn_image5.html](#) → hint [[exam](#)]

70



Hint

wk13-실습 : 결과를 나의 github에 올리기

71

실습 결과를 github에 올립니다.

1. README.md에는 실습 결과 요약 추가 입력
2. hmxx_image5.html 완성
3. “hmxx” repo의 wk13 폴더에 upload

단 업로드가 안될 경우, wk13.zip을 업로드
그리고 집에서 wk13 폴더로 다시 업로드.

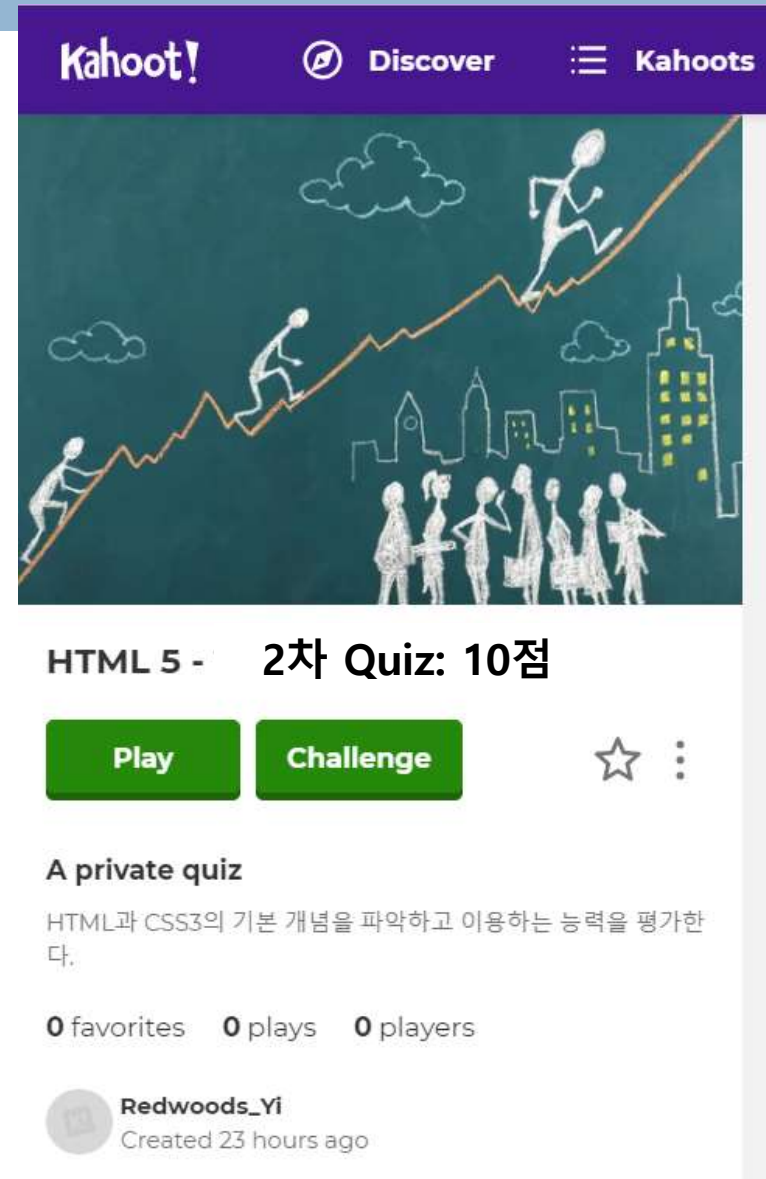
예비 기말고사: Kahoot Quiz 10

72

일시 : 2020. 6. 3 (목) 6 교시

장소 : E323

배점 : 10 %



The image shows a screenshot of a Kahoot! quiz page. At the top, there is a purple header with the Kahoot! logo, a 'Discover' button, and a 'Kahoots' button. Below the header is a large illustration of a city skyline at night with a yellow line graph trending upwards. Three stick figures are running along the line graph, and a group of stick figures is standing at the bottom. Below the illustration, the quiz title 'HTML 5 - 2차 Quiz: 10점' is displayed. There are two green buttons labeled 'Play' and 'Challenge', followed by a star icon and a vertical ellipsis. Below these buttons, the text 'A private quiz' is shown, followed by a description: 'HTML과 CSS3의 기본 개념을 파악하고 이용하는 능력을 평가한다.' At the bottom, it says '0 favorites 0 plays 0 players' and shows the creator's name 'Redwoods_Yi' with a profile picture and the text 'Created 23 hours ago'.

Kahoot! Discover Kahoots

HTML 5 - 2차 Quiz: 10점

Play Challenge ☆ ⋮

A private quiz

HTML과 CSS3의 기본 개념을 파악하고 이용하는 능력을 평가한다.

0 favorites 0 plays 0 players

Redwoods_Yi
Created 23 hours ago

Project-3. hmxx_rpt03.html

일시 : 2020. 6. 3 (목) ~ 6.20 (일)

장소 : 집, 카페, ...

내용 : My Final Homepage V3 (10%)

- ① Semantic tag로 화면 layout 구성 (가로 또는 세로 메뉴 구성)
- ② 타이틀 : hmxx : 나는 누구인가?
- ③ 나의 하루
- ④ My Best 3 (책, 영화, 음악, 그림, 여행 등에서 **선택 1**)
- ⑤ 동호회 모집 form.html 추가 (Web form, fieldset tag는 반드시 사용)
- ⑥ X-MAS파티 초대장 (기 제작된 X-MAS 카드 활용)
- ⑦ Html, css3, javascript를 활용하는 능력을 평가.
- ⑧ 적절한 그림, 음악, 동영상은 가능한 link로 처리해서 연결하고, 필요 시 media folde에 저장해서 이용.

- hmxx_rpt03 폴더를 github에 만든다. (단 hmxx는 본인 id로 변경!)
- hmxx_rpt03 폴더에 README.md 파일을 만들어서 홈페이지에 대한 설명을 추가한다.
- hmxx_rpt03 폴더에 나의 홈페이지에 사용되는 모든 파일들을 올린다.
- 나의 홈페이지 시작 파일은 hmxx_rpt03.html

기말고사(필기) 안내

74

일시 : 2020. 6. 10 (목) 6 교시(오후 2시)

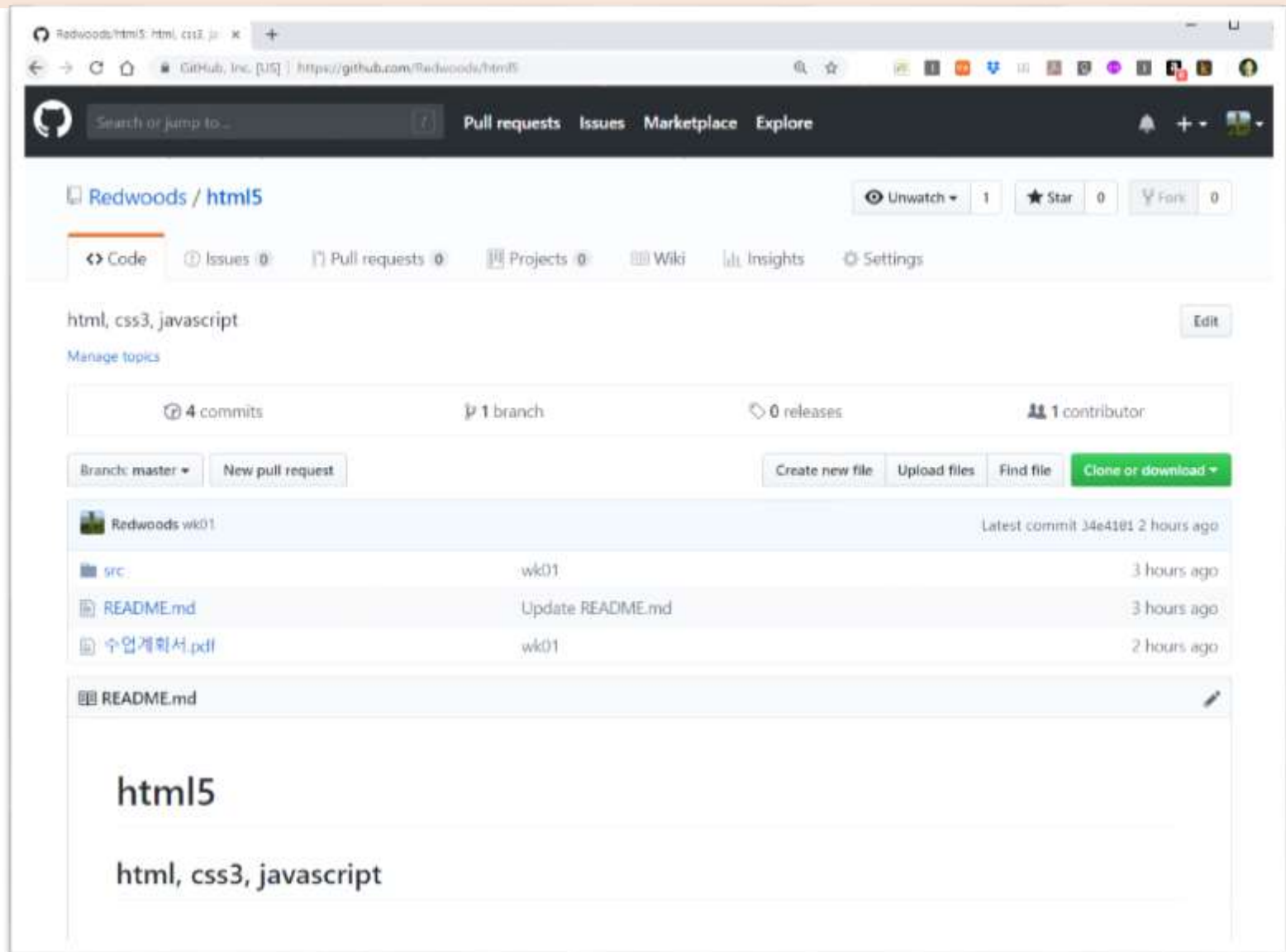
장소 : ? (추후 공지)

시험범위 : HTML5 & CSS3 (교재 6장 ~ 10장) --- 10점 (10%)

- 교재 연습문제/Kahoot-Quiz (이론문제)에서 30% 출제
- 실습에서 배운 html code에서 70% 출제 (객관식, 단답형)

HTML5 : 강의자료실

<https://github.com/redwoods/html5>



교재 WEB 강의 소개

← → ↻ ① webprogramming.co.kr ☆



HTML5+
CSS3+
Javascript

웹 프로그래밍

Home Introduction Notice Board Support Code



HTML5 +
CSS3 +
Javascript

웹 프로그래밍

HTML5로
여러분의 무한한
상상력을
표현해 보세요!



Sir Tim Berners-Lee
(1955.6.8 ~)

명품 웹 프로그래밍 소개
"웹 프로그래밍을
가장 쉽게 익힐 수 있는 책"

처음 웹 프로그래밍을 공부하는 입문자들도
모든 주제를 직관적으로 이해하고
빠르게 파악할 수 있습니다.



자세히보기 →

강력한 Q&A 피드백 제공
"빠르고, 간결하고, 정확한
저자의 직접적인 답변"

'이거 이해가 잘 안되는데.. 물어볼 사람도 없고..'
더이상 고민하지 마세요.
명품 웹 프로그래밍 홈페이지에서는
누구나 저자가 직접 답변해주는
Q&A 게시판들을 이용하실 수 있습니다.



자세히보기 →

즉석 실행 가능한 예제 프로그램
"백문이 불여일견, 백견이 불여일타(打)!"

코드뿐만 실행되어 있는 예제들.
결과 화면이 있어도 이해가 잘 안되시죠?
예제 소스를 바탕으로, 내맘대로 수정한
코드를 즉석으로 웹 페이지로
변환해주는 예제 프로그램을 통해
모든 코드를 빠르고 쉽게
이해할 수 있습니다.



자세히보기 →

Notice

Test

2017-01-16 15:32

Know-How

Test

2017-01-17 14:04 관리자

76

<http://webprogramming.co.kr>

관련 WEB 강의 소개 – w3schools.com

The screenshot shows the w3schools.com website. The browser address bar displays "https://www.w3schools.com". The website has a green header with the logo "w3schools.com" and the tagline "THE WORLD'S LARGEST WEB DEVELOPER SITE". Below the header is a green navigation bar with links for "TUTORIALS", "REFERENCES", and "EXAMPLES".

On the left side, there is a sidebar menu with the following categories and links:

- HTML and CSS
 - Learn HTML
 - Learn CSS
 - Learn W3.CSS
 - Learn Colors
 - Learn Bootstrap
 - Learn Icons
 - Learn Graphics
 - Learn How To
- JavaScript
 - Learn JavaScript
 - Learn W3.JS
 - Learn JQuery
 - Learn JQueryMobile
 - Learn AppML
 - Learn AngularJS
 - Learn JSON
 - Learn AJAX
- Server Side
 - Learn SQL
 - Learn PHP
 - Learn ASP
- Web Building
 - Web Templates
 - Web Statistics
 - Web Certificates
- XML Tutorials
 - Learn XML
 - Learn XML AJAX
 - Learn XML DOM
 - Learn XML DTD
 - Learn XML Schema
 - Learn XSLT
 - Learn XPath
 - Learn XQuery

The main content area is divided into three sections:

- HTML**: The language for building web pages. It includes a "LEARN HTML" button and an "HTML REFERENCE" button. An "HTML Example:" box shows a code snippet for a basic HTML document structure.
- CSS**: The language for styling web pages. It includes a "LEARN CSS" button and a "CSS REFERENCE" button. A "CSS Example:" box shows a code snippet for styling a body and a div.
- JavaScript**: The language for programming web pages. It includes a "JavaScript Example:" box showing a code snippet for a function that gets an element by ID and changes its style.